UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/997,404	11/29/2001	Padmanabhan Sreenivasan	499.074US2	3328

21186 7590 07/26/2010

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER	
BRUCKART, BENJAMIN R	

ART UNIT	PAPER NUMBER
2446	

NOTIFICATION DATE	DELIVERY MODE
07/26/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

uspto@slwip.com
request@slwip.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte PADMANABHAN SREENIVASAN, AJIT DANDAPANI, MICHAEL NISHIMOTO, IRA PRAMANICK, MANISH VERMA, ROBERT DAVID BRADSHAW, LUCA CASTELLANO, and RAGHU MALLENA

Appeal 2009-006251[1]
Application 09/997,404
Technology Center 2400

Before JOHN A JEFFERY, JEAN R. HOMERE, and JAMES R. HUGHES, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL[2]

---

[1] Filed November 29, 2001. The real party in interest is Silicon Graphics, Inc. (App. Br. 3.)

[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the "MAIL DATE" (paper delivery mode) or the "NOTIFICATION DATE" (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) (2002) from the Examiner's final rejection of claims 1, 2, 4, 5, 7 through 12 and 14 through 19. Claims 3, 6, and 13 have been canceled. (App. Br. 5.) We have jurisdiction under 35 U.S.C. § 6(b) (2008).

We affirm.

Appellants' Invention

Appellants invented a highly available (HA) system for implementing a failover policy that redistributes the current network load between the functioning nodes thereof upon detecting that one or more of the network nodes has failed. (Spec. 2, ll. 17-21.) As depicted in Appellants' Figure 1A, the HA system (10) includes a plurality of servers or nodes (20), each clustering a plurality of processors, and each being coupled to a database (26) and one more client computers (24) via a network (22). (*Id.* at. 10, ll. 22-29.) As shown in Figure 1B, each of the nodes (20) includes an HA base software (16) that monitors the operation of the network nodes (20). (*Id*. at 11, ll. 35-38.) Upon detecting that a network node (20) has failed, the HA base software (16) executes a failover script that utilizes the initial number of nodes (20) within a same cluster to which a resource group (initial failover domain) is allocated to generate a current list of network nodes within the same cluster assigned to the group resource (runtime failover domain). (*Id*. at 9, ll. 7-12, *id*. 14, ll. 3-9.) The failover script subsequently uses the runtime failover domain along with the failover attributes to rebalance the network load by reassigning the resource groups of the failed

node to another node in the runtime failover domain. (*Id.* at 9, ll. 13-15, *id* at 15, ll. 10-14.)

Illustrative Claim

Independent claim 1 further illustrates the invention. It reads as follows:

1. A system for implementing a failover policy comprising:

a cluster infrastructure for managing a plurality of nodes;

a high availability infrastructure for providing group and cluster membership services; and

a high availability script execution component operative upon the detection of a failover event to perform the tasks of:

receive a failover script comprising a set of one or more commands and further operable to receive at least one failover attribute and operative to cause the failover script to be interpreted to produce a run-time failover domain from an initial failover domain, and

execute one or more action scripts, the action scripts when executed causing a resource group to failover to a node in the run-time failover domain, the resource group having one or more resources.

Prior Art Relied Upon

The Examiner relies on the following prior art as evidence of unpatentability:

Le	US 6,145,089	Nov. 7, 2000
Alexander	US 6,189,111 B1	Feb. 13, 2001
Chao	US 6,438,705 B1	Aug. 20, 2002

Microsoft Computer Dictionary, Fifth Edition; Microsoft Press; Copyright 2002[3]

Rejection on Appeal

The Examiner rejects claims 1, 2, 4, 5, 7 through 12 and 14 through 19 under 35 U.S.C. § 103(a) as being unpatentable over the combination of Alexander, Le, and Chao.

Appellants' Contentions

Appellants contend that the combination of Alexander, Le, and Chao does not teach or suggest a failover script interpreted to produce a runtime failover domain from an initial failover domain, as recited in independent claim 1. (App. Br. 12-13.) According to Appellants, Alexander discloses a routine execution mechanism that executes a routine to remove a detected failing node from a bitmap of node cluster, whereas the claim requires a failover script that produces a runtime failover domain. (*Id.*) Similarly, Appellants argue that while Le discloses action scripts that can start or stop services within a service group or domain, they differ from failover scripts that determine which nodes are within the failover domain. (*Id.* at 13.) Additionally, Appellants argue that Chao does not remedy the noted deficiencies of the cited references. (*Id.*)

[3] We note that the effective date of this dictionary is after the filing date of the present application. However, in this Opinion, we rely upon a dictionary definition with an effective date of 1994 to properly construe the disputed terms in this appeal. *See infra* n.3.

Examiner's Findings

The Examiner finds that while Alexander does not particularly disclose a failover script, its disclosure of an execution mechanism for investigating an error and for taking action is also capable of providing a platform for the failover script. (Ans. 10-11.) Further, the Examiner finds that Le complements Alexander's teachings by disclosing a script file (450) that provides a response to a service failure, which the Examiner construes as a failover script. (*Id*. at 12.) Therefore the Examiner concludes that the proffered combination teaches the disputed limitations. (*Id*.)

II. ISSUE

Have Appellants shown that the Examiner erred in concluding that the combination of Alexander, Le, and Chao teaches or suggests a failover script that produces a runtime failover domain from an initial failover domain, as recited in independent claim 1?

III. FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

Appellants' Specification

1. Appellants' Specification discloses that the network administrator configures the failover policy to include a failover domain, failover attributes and a failover script. (Spec. 8, l. 26-27, *id.* at 9, l. 1-3.)

2. The failover domain is a list of ordered nodes within a same cluster on which a given resource group can be allocated. (*Id.* 9, ll. 7-9.)

3. Appellants' Specification defines a failover script as a script shell that generates a runtime failover domain and returns it to the High Availability base software process. (*Id.*, ll. 25-26.)

4. The administrator defines an initial failover domain when creating the failover policy. Upon detecting a failure, the failover script converts the initial failover domain to a runtime failover domain to thereby update the list of available nodes on which a resource group should reside. (*Id.*, ll. 11-15.)

Alexander

5. Alexander discloses a scalable and fault tolerant system for enhancing the survivability of network components in the event of a catastrophic failure. (Abst.)

6. As depicted in Figure 1, Alexander discloses a plurality of processor nodes (12), each running a cluster management service to independent access the memory of other nodes, as well as to periodically send "I'm alive" messages to the other nodes. (Col. 5, ll. 20-22, ll. 40-45.) Further, each of the nodes in Alexander's high availability system includes a processor interface (14) that enables a user to directly interact with resources stored at each of the nodes. (Col. 5, ll. 15-19.)

7. Upon detecting that a node fails to provide a status message, the cluster management component at the other non-failed nodes update their view of the cluster by deleting the failed node therefrom, and harvesting the

resources of the failed node to another non-failed node. In particular, Alexander discloses using an execution routine to investigate the failure and to take remedial actions, including redistributing the resources of the failed node to the other non-failed nodes. (Col. 5, ll. 46-50, col. 6, ll. 36-49, col. 8, ll. 37-44.)

Lee

8. As shown in Figures 1 and 4, Lee discloses a server failover system having a plurality of servers (110, 120, 130), each including a service manager (420) for providing a plurality of services. (Col. 2, ll. 22-29.)

9. Each of the servers further includes a role manager (410) that periodically receives heartbeat messages from the other servers. Upon failing to receive a status message from one of the servers, the role manager elects another server to perform the service (s) failed server. In particular, the service manager (420) of the elected server runs script files (450) of the service (s) to acquire and release them. The script files may include scripts to start, stop, and restart one or more services, as well a response to a service failure. (Col. 4, ll. 15-46.)

IV. ANALYSIS

Claims 1, 2

Independent claim 1 requires, *inter alia*, a failover script that produces a runtime failover domain from an initial failover domain. (Br. 17, Claims App'x.)

We first consider the scope and meaning of the cited claim limitation, which must be given the broadest reasonable interpretation consistent with Appellants' disclosure, as explained in *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997):

> [T]he PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification.
>
> *Id.* at 1054. *See also In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989) (stating that "claims must be interpreted as broadly as their terms reasonably allow.").

Appellants' Specification defines a failover script as a script that converts an initial failover domain to generate a runtime failover domain upon detecting a node failure. (FF. 3-4.) Further, the Specification defines a failover domain as a list of ordered nodes within same cluster to which resources can be allocated. (FF. 2.)

Our reviewing court further states, "the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan after reading the entire patent." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1321 (Fed. Cir. 2005) (en banc).

Upon reviewing Appellants' Specification, we find the claim recitation of a failover script that produces a runtime failover domain from an initial failover domain can be broadly, but reasonably construed as a

script[4] that, upon detecting a node failure, modifies the initial list of ordered nodes to delete the failed node therefrom, and to distribute the service of the failed node to a node in the list of non-failed nodes.

As set forth in the Findings of Fact section, Alexander discloses an execution routine that, upon detecting a node failure, updates the view grid of the non-failed nodes to delete the failed node therefrom, and to subsequently reassign the services of the failed node to a non-failed node. (FF. 7.) Further, Lee discloses a role manager that, upon failing to receive a status message from a server, executes a script file that reassigns the services of a failed server to an elected non-failed server. (FF. 9.) Consistent with our claim interpretation above, we find that by using a routine to modify the grid of non-failed nodes to delete the failed node therefrom, Alexander teaches or at least suggests a failover routine that converts an initial failover domain (the initial node grid) to a runtime domain (the updated node grid) upon detecting a node failure. Further, we find that an ordinarily skilled artisan would have readily appreciated that Lee's disclosure of using a script to reassign the services of a failed server node to a non-failed server node reinforces Alexander's notion of using a routine to update the grid of non-

[4] We note that Appellants' Specification does not provide a definition for a script. Thus, we turn to a computer dictionary for the ordinary meaning of 'script,' which is defined as a type of program consisting of a set of instructions to an application or utility program. Microsoft Computer Dictionary, 350, 2d ed. 1994. Consequently, we find the Examiner's construction of a script as comprising at least a routine or any portion of a computer program to be reasonable because the program consists of a set of instructions. (Ans. 11.)

failed nodes. Therefore, we find that the reference combination of Alexander and Lee discloses known elements that perform their ordinary functions to predictably result in a highly available system whereupon detecting a failed node, a script or a set of instructions are used to delete the failed node from an initial grid of non-failed node, and to thereby generate an updated grid of non-failed node.

Regarding Appellants' argument that Chao does not cure the noted deficiencies of Alexander and Lee (Br. 13,) we find no such deficiencies in the cited references for Chao to remedy. It follows that Appellants have not shown that the Examiner erred[5] in concluding that the combination of Alexander, Lee, and Chao renders independent claim 1 unpatentable. Similarly, claim 2 falls with claim 1 since they are argued as a single group. 37 C.F.R. § 41.37(c)(1)(vii).

Claims 7-12 and 14-19

Appellants argue that Alexander's ASSERT macros are used for performing ad-hoc consistency checks, but they do not consist of an action script for verifying that a resource is configured on a target node, as recited

[5] Appellants have the burden on appeal to the Board to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) ("On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence of secondary indicia of nonobviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

in claim 7. (Br. 14.) In response, the Examiner finds that Alexander's disclosure of an application that uses techniques of an operating system to validate its own data structure teaches the cited limitation. (Ans. 13.) We agree with the Examiner because each node's validation of its own data structure also entails managing its current view of the network including the data structure of other nodes including a target node. Moreover, since Appellants have failed to file a Reply Brief to rebut the cited finding made by the Examiner, we find that Appellants have therefore failed to show error in the Examiner's rejection of claim 7. Similarly, claims 8 through 12 and 14 through 19 fall with claim 7 since they are argued as a single group. 37 C.F.R. § 41.37(c)(1)(vii).

Claim 19

Appellants argue that Alexander does not teach or suggest an application plug-in that provides a high availability interface for the application, as recited in claim 19. (Br. 15.) In response, the Examiner finds that Alexander's disclosure of an interface utilized to communicate with a user of client application teaches the cited limitation. (Ans. 10.) We agree with the Examiner. Alexander's high availability system discloses that each of the nodes therein includes a processor interface that enables a user to directly interact with resources stored at each of the nodes. (FF. 6.) We find that since a plug-in can be reasonably but broadly construed as an interface, the disclosed processor interface teaches or at least suggests the claim limitation.

Claims 4, 5

Appellants have not argued or even attempted to traverse the rejection of claims 4 and 5. Therefore such arguments are waived. We have considered in this decision only those arguments Appellants actually raised in the Briefs. Any other arguments which Appellants could have made but chose not to make in the Briefs are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(vii).

V. CONCLUSION OF LAW

Appellants have not established that the Examiner erred in rejecting under 35 U.S.C. § 103(a) claims 1, 2, 4, 5, 7 through 12 and 14 through 19 as set forth above.

VI. DECISION

We affirm the Examiner's rejection of claims 1, 2, 4, 5, 7 through 12 and 14 through 19.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv) (2009).

AFFIRMED

pgc

Appeal 2009-006251
Application 09/997,404

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402